COP 290 Assignment 1

Moodle App

This assignment involves developing a frontend for the moodle android app. We'll provide you with a web2py server which you can locally deploy. You are supposed to code an android app which will interact with this server via API calls. The entire app design has to be done by you.

What is Web2Py?

web2py is defined as a free, open-source web framework for agile development which involves database-driven web applications; it is written in Python and programmable in Python. It is a full-stack framework; it consists of all the necessary components, a developer needs to build a fully functional web application.
web2py framework follows the **Model-View-Controller** pattern of running web applications unlike traditional patterns.

- **Model** is a part of the application that includes logic for the data. The objects in model are used for retrieving and storing the data from the database.
- **View** is a part of the application, which helps in rendering the display of data to end users. The display of data is fetched from Model.
- **Controller** is a part of the application, which handles user interaction. Controllers can read data from a view, control user input, and send input data to the specific model.

- **web2py** has an in-built feature to manage cookies and sessions. After committing a transaction (in terms of SQL), the session is also stored simultaneously.
- **web2py** has the capacity of running the tasks in scheduled intervals after the completion of certain actions. This can be achieved with **CRON** (time-based job scheduler in Unix-like computer operating systems)

http://www.tutorialspoint.com/web2py/index.htm

App Features

Your App should have following features . For every feature required you will be provided with corresponding server API that you need to call :

- Login (as a student or faculty)

After Login the user should be able to :

- View the list of courses that a student is registered for(or the list of courses the faculty is taking)
- View all notifications (related to threads)
- View grades for all courses
- View Information of a particular Assignment
- View the list of all assignments for a particular course
- View grades for a particular course
- View the list of all threads for a particular course
- View the information of a particular thread
- Create a new thread for a particular course
- Add a new comment to an existing thread

- Logout

Setting up

Pre-requisites: For this assignment, we have provided you a server which will run on a Linux machine. It's a web2py (python based) server so it's imperative that you install python on your ubuntu machine.
Also install **python-tk** package.
Install JSON viewer addon in the browser you are using. It will make things simpler when you are trying to understand the output json format of the APIs.

After installing the prerequisite packages, on the terminal, clone the server code from the bitbucket repo as follows:

```
git clone https://jas92@bitbucket.org/jas92/moodle-plus.git
```

First **cd** into the project folder you have just cloned, then run the following command:
```
python web2py.py
```
This will open a GUI (only if python-tk package is installed). Specify any port (default is 8000) and password and start the server.

As soon as you'll start the server, your default browser window will open the MoodlePlus homepage. At this stage, the databases are empty so you'll not be able to login.

The code includes a file to initially populate the databases with some dummy data. Open another terminal and from the server folder, run the following command to populate the databases.
```
python web2py.py -M -S MoodlePlus -R applications/MoodlePlus/populate_db.py
```

You can login now as any of the following users:

Students
John Doe
username: cs1110200
password: john

Jasmeet Singh
username: cs5110281
password: jasmeet

Abhishek Bansal
username: cs5110271
password: abhishek

Shubham Jindal
username: cs5110300
password: shubham

Professors
Prof. Vinay Ribeiro
username: vinay
password: vinay

Prof. S.C Gupta
username: scgupta
password: scgupta

Prof. Subodh Kumar
username: subodh
password: subodh

The students and professors have been registered with some courses.

Populating the databases further
After logging in as a student, choose a course (say design practices) and go to the threads tab. Add some threads and comments.

Server APIs

The following is the list of APIs that you'll use.
Login: `/default/login.json?userid=<username>&password=<password>`
Logout: `/default/logout.json`
List of courses registered: `/courses/list.json`
All notifications: `/default/notifications.json`
All grades: `/default/grades.json`
list of all assignments: `/courses/course.json/<coursecode>/assignments`
info of a particular assignment: `/courses/assignment.json/<assignment-number>`
Course grades: `/courses/course.json/<coursecode>/grades`
List of course threads: `/courses/course.json/<coursecode>/threads`
info of a particular thread (including all comments):
`/threads/thread.json/<thread-number>`
Create new thread:
`/threads/new.json?title=<title>&description=<desc>&course_code=<coursecode>`
Add comment to a thread:
`/threads/post_comment.json?thread_id=<thread_id>&description=<desc>`

NOTE: coursecode in above endpoints should be in lowercase characters.

You are required to make use of each of the above endpoints in your app.

Server requests

You'll have to make GET requests on the above end points and you'll receive json response which you need to parse. Get familiar with the json response formats of all the above mentioned endpoints by typing the following in your browser:
localhost:<port_number> followed by the above mentioned URL extensions.
For example, **`localhost:8000/courses/list.json`**

The JSON viewer addon will display the json in a nice, easy-to-read format.

Note that arguments to the GET request are passed in the URL itself, for example, the end point for login:
`localhost:8000/default/login.json?userid=cs1110200&password=john`

For making a GET request from your android app, replace **localhost** with the IP address of the machine on which server is running, for example,
`192.168.1.2:8000/courses/list.json`
Make sure that your machine and phone are both within the same network, for example, both are connected to IITD network.

Grading

The grading for each assignment will be done on a scale of 100. For this assignment, the breakup of the grade is:

1. Design document [20 marks]: You should submit the design of the application in the latex format. The design should highlight the basic user interface, the high level functions used in the code, and the possible error scenarios that the application handles.
2. Basic code and completeness [40 marks]: This should contain the source code of your application.
3. Software practices
    a. Code indentation [5 marks]: A badly indented code is difficult to maintain as well as debug. Use a proper editor to write your code.
    b. Commenting [5 marks]: We expect at least one meaningful comment describing each function and each source file.
    c. Modularity [5 marks]: All the big modules of the code should be maintained in separate source files. For example, if you are developing a

Stack data structure using a Linked list, you are focussing on two separate modules; the stack and the linked list. This should be maintained in two different files.

   d. Version control system [5 marks]: Create a repository to maintain your code. We advise you to use either of this two options: bitbucket.org, and github.com. Use this repository to exchange code between team partners, and to create multiple revisions of the code. At the time of demo, we will check the commits of all the team members, and if we found that a team member is not contributing to the submission, he/she will receive 0 marks for the assignment

4. Usability of the application [20 marks]: Application should be easy to use and should have a nice interface

Submission

You should submit the assignment on Moodle. The assignment should be bundled in a zip file whose name is entryno1_entryno2_entryno3.zip. The zip must contain three folders:

1. src: This contains the entire source of your Android application
2. apk: This contains the apk file which will run on the mobile
3. doc: This contains a file main.tex that contains the design of the application.

If your submission does not adhere to the given submission format, you will receive a penalty of 10 marks.

At the time of demo, you should bring your laptop and an Android based mobile on which this application can run. We will use your Moodle submission to create an apk file and we will evaluate your assignment for this apk file

Academic Integrity

All the submissions will be checked for plagiarism. If a student group is found to be copying code from other group or a third party source, all the concerned students (including the students who made their code available to other groups) will get ‘F’ grade and a disciplinary action will be initiated against them. You can use the code obtained from Android examples on the Internet; however, while doing so, you must clearly mention the source of the code in comments.